

特許願 13

昭和49年 7月17日

特許庁長官殿

発明の名称

デジタル計算機

発明者

東京都千代田区丸の内一丁目280番地  
株式会社日立製作所中央研究所内  
〒100 日立製作所

特許出願人

東京都千代田区丸の内一丁目5番1号  
株式会社日立製作所  
〒100 日立製作所

代理人

東京都千代田区丸の内一丁目5番1号  
株式会社日立製作所内  
電話東京 270-2111 (大代表)  
(7237) 弁護士 根田利雄

⑪特開昭 51-10746

⑬公開日 昭51.(1976) 1.28

②特願昭 49-81108

②出願日 昭49.(1974) 7.17

審査請求 未請求 (全4頁)

庁内整理番号 6522 56

6341 56  
6450 56  
6619 56

⑫日本分類

770F2  
770E1  
770H1  
770F8

⑬Int. Cl?

G06F 7/06  
G06F 7/00  
G06F 15/06

明 細 書

発明の名称 デジタル計算機

特許請求の範囲

命令語を処理するデジタル計算機において、命令語の演算を実行する命令実行ユニットが複数個あり、それらが各々別々に命令語の処理可能とした時、各々の命令実行ユニットにおける演算結果を命令語の順序どおりに処理するため、それらの演算結果を貯えておくキューレジスタと命令語を順序どおりに処理するための演算実行管理テーブルとそれらのレジスタとテーブルを制御する順次実行制御装置を具備することを特徴とするデジタル計算機。

発明の詳細な説明

本発明は、命令語を処理するデジタル計算機に関するものである。

一般にデジタル計算機の構成は第1図のようになつてゐる。第1図において1は主記憶装置、2は命令制御ユニット、3は記憶装置制御ユニット、4は命令実行ユニットである。本発明はこれ

らの内、4に関するものである。

従来、命令語を高速に処理する方式として、パイプライン制御方式が知られている。この方式は命令の各実行段階を、別々の論理装置(ユニット)に分担させ一種の流れ作業を実施している。しかし、命令語の種類が幾つかあり、それらの命令語の処理時間は一定ではない。そのため、パイプライン制御方式のメリットがそこなわれることが多

多ある。そこで、命令制御ユニットでは、解読した命令語の処理時間を予測して、順次命令語の解読を行ない、命令実行ユニットに制御を移す時に、命令語によつて、その命令語に適した命令実行ユニットに移す。これは、本発明の対象外なので説明は省く。このように、命令実行ユニットを複数個もつことにより、命令語の処理を早めようとするものである。しかし、命令語は並んでいる順番通りに実行されるべきものであるため、このような制御方式を採用した場合、以前に解読した命令語より後で解読した命令語の処理の方が、先に終了する

場合がある（命令語により処理時間が一定でないため）。ところが、ここで命令語の処理の終了とは、汎用レジスタや、浮動小数点レジスタ、記憶装置への書き込みが終了した時点でのことである。そこで、本発明はこれらのレジスタや、記憶装置に書き込む以前の処理（演算器等での演算処理）が終了した時点で、その結果を一時貯えておくことにより、最後の書き込み処理だけは、命令語の並んでいる順番通りに行なおうとするものである。

以下、図面により詳細に説明する。

第3図は、本発明を適用した場合の命令実行ユニットのブロックダイアグラムである。この図において、5、6、7は命令の演算を実行する装置（加算器、シフト、乗算器、除算器等）である。この図は例として3つの演算装置（5、6、7）がある場合であるが、一般的には複数個である。5、6、7で演算実行が終了した時点で結果をキューレジスタ8に移す。キューレジスタ8と演算実行管理テーブル11のフォーマットを第2図に示す。8は11で指定されたキューレジスタ番号

群の中のうち、該当するキューレジスタ番号を選択してそれに対応する連続番号（連続番号表示子）と演算が完了したか否かを示す演算完了表示ビット（演算が完了していれば“1”、していない場合は“0”）と5または6または7で演算された結果の入るレジスタ（演算結果レジスタ）よりなり、また11は、連続番号（1～n：nは固定値）とその連続番号で示されるキューレジスタ番号の内容が有効か否かを示す有効ビットと、キューレジスタ番号よりなる。第2図では、8、11の内容の一例を示してある。つまり、8の中には、11のキューレジスタ番号のうちQと表示されたものの連続番号がとりこまれている。また、連続番号1だけは演算が終了し、演算結果レジスタにセットされていることを示す。

第3図において、命令制御ユニット2より実行すべき命令をどの演算装置で実行させるかを示す番号（つまり、キューレジスタ番号と等しいもの）がL1を通して、11にセットされる。11の中でどこへセットするかは、L2の指示により

（つまり有効ビット）連続番号記憶レジスタ15（第4図）の内容より大きい数字で最初に有効ビットが“0”のところを探し出して、そこにキューレジスタ番号をセットすると同時にその有効ビットも“1”にする。11の内容は、L5、L21、L22を通して、順次実行制御装置10に送られて、8の読出し、11の更新の制御をする。5、6、7の演算結果はL12、L14、L16を通して8に送られる。その際L11、L13、L15を“1”とする。8の中のどこにセットするかはL9の制御による。L9のキューレジスタ番号と、連続番号の制御により8にセットする。L6、L7、L8の状態をみて10で制御を行なつてL10の制御により、L17、L18、L19のうち1本を選択してL20を通して、汎用レジスタ、浮動小数点レジスタ、記憶装置書き込みレジスタ等(9)にセットする。

第4図は、順次実行制御装置10の詳細図である。この図を使つて10の役割を説明する。この図において、12、13、14、18、19、

20、22、23はアンドゲート、21はオアゲート、15は連続番号記憶レジスタ、16は15の内容を更新（+1）する回路、17は遅延線である。L21、L22を通じて、連続番号、有効ビットが送られ、12でアンドが成立した時L23がオンとなり、L5を通してキューレジスタ番号が送られ、13によりアンドが成立した時、L9がオンとなる。一方、14によりL9、L21、L27のアンドがとれた時L10がオンとなり、8より読出しを行なう。その際、22、23により、連続番号（16により更新された値を15より読出す）とキューレジスタ番号をL3、L4を通して、11に送る。また14のアンドが成立しない時は、L10はオフとなり16の起動をかわないでいく。またその時は、L28がオンとなり17により1サイクル遅らせて、L29をオンとすることにより12の制御を行なう。また一方、L6、L7、L8はL5により制御して、18、19、20のうち1つのアンド条件が成立して、L24、L25、L26の内1つがオンとなり、

21を通してL27が制御される。これは、演算が終了しているか否かをチェックするためのものである。

このような装置を具備することにより、複数の命令実行ユニットをもつ計算機において、演算結果を独立に順次処理することにより、命令語の高速処理が期待される。

図面の簡単な説明

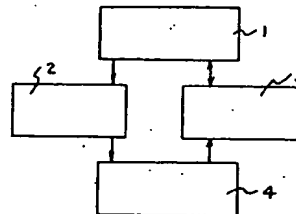
第1図は、一般的なデジタル計算機の構成、第2図は、本発明に特徴的なキューレジスタと演算実行管理テーブルのフォーマットを示す図、第3図は、本発明を適用した場合の命令実行ユニットのブロックダイアグラム、第4図は本発明に特徴的な順次実行制御装置の図である。

1：主記憶装置、2：命令制御ユニット、3：記憶装置制御ユニット、4：命令実行ユニット、8：キューレジスタ、11：演算実行管理テーブル、10：順次実行制御装置

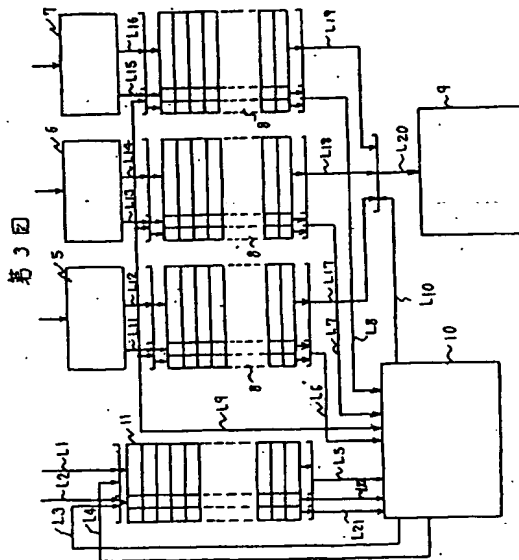
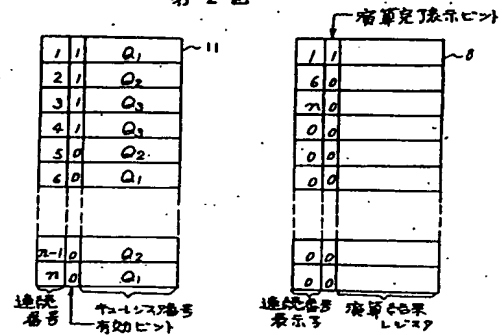
代理人 弁理士 藤田利幸

特開 昭51-10746 (3)

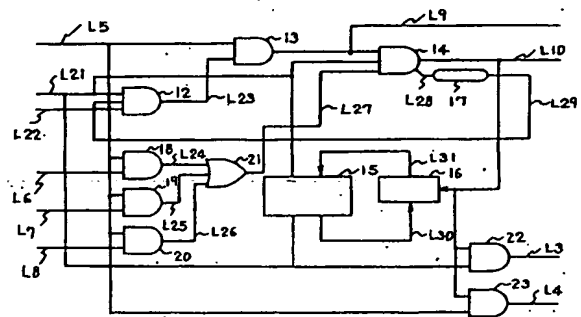
第1図



第2図



第4図



## 添附書類の目録

- (1) 明 書 1 通  
 (2) 図 面 1 通  
 (3) 考 査 証 1 通  
 (4) 特 許 証 1 通

特開 昭51-10746 (4)

前記以外の発明者、特許出願人または代理人

発 明 者

東京都国分寺市東恋ヶ橋1丁目280番地  
 株式会社 日立製作所中央研究所内  
 河 辺 俊

Kokai No.: S51-10746

Publication date: January 28, 1976

Application No.: S49-81109

Application date: July 17, 1974

Inventor: Shimoi Kenji

Applicant: Central Research Laboratory, Hitachi Ltd.

### Specification

[Title of the Invention]

Digital Computer

[Claims]

In a digital computer which processes instruction word, a digital computer comprises;

plural of instruction execution units to execute computing of instruction word,

when each of plural of instruction execution units can execute an instruction word respectively,

a queue register for storing computing results for processing a computing result at each instruction execution unit in an instruction words order,

a computing execution control table to process instruction words in the order,

a sequential execution control device for controlling the table and the register.

[Detailed description of the Invention]

The present invention is related to a digital computer to process an instruction word.

Generally, a structure of digital computer is as shown in Fig.1. In Fig.1, numeral 1 denotes a main memory, numeral 2 denotes an instruction control unit, numeral 3 denotes a memory device control unit, and numeral 4 denotes an instruction execution unit. The present invention is related to the instruction execution unit denoted by numeral 4.

In the past, a pipe-line control method has been known as a method to process an instruction words at a high speed. In this method, each execution

step of an instruction is assigned to different logical units to perform a kind of assembly-line. However, there are several types of instruction word, and processing time of each instruction word is different from each other. Then, merits of the pipe-line control method are often reduced.

Consequently, in the instruction control unit, sequential instruction words are decoded with estimating a processing time of a decoded instruction word, and when the control is shifted to the instruction execution unit, an instruction word is sent to the most appropriate instruction unit for the instruction word. As that is out of the scope of the present invention, explanation is omitted. As mentioned above, this is a method to speed up processing of instruction words with placing plural of instruction execution units. However, as instruction words should be executed in a order of arrangement, when this method is applied, processing of an instruction word decoded later is sometimes completed before processing of an instruction word decoded earlier (as a processing time varies according to an instruction word). Wherein, completing of processing of an instruction words means completing of writing in a general register, a floating point register and a storage device. Then, the present invention performs the last writing process in the order of arrangement of instruction words with storing results temporally when a process before the writing process (computing with a computing unit and so on) is completed.

Hereinafter, the present invention is explained in detail with drawings.

Fig.3 is a block diagram of an instruction execution unit to which the present invention is applied. In the figure, numerals 5, 6, 7 denote devices executing computing of instructions (an accumulator, a shifter, a multiplier, a divider and so on). Though this drawing shows a case of three computing units, generally plural of computing units are placed. When execution of computing is completed at 5, 6, 7, results are sent to a queue register 8. A format of queue register 8 and a computing execution control table 11 is shown in Fig.2. A part denoted by numeral 8 includes a sequential number (sequential number displayer) corresponding to a selected queue register number among a group of queue register numbers set by 11, a computing completion bit (when computing has been completed "1", when computing has not been completed "0"), and a register (computing result register) to store a result computed by 5, 6 or 7, a part denoted by numeral 11 includes a sequential number (1 to n: n is a fixed value), a valid bit to indicate whether contents of

a queue register number are valid, and a queue register number. Fig.2 shows an example of contents of 8 and 11. That is, a sequential number of a queue register number only displayed as  $Q_2$  among queue register numbers of 11 is taken into 8. And also, it is shown that computing of only sequential number 1 has been completed and a result is set into the computing result register.

In Fig.3, a number to indicate what computing unit executes an instruction to be executed (that is, an equivalent number to the queue register number) is set by an instruction control unit 2 to 11 through  $L_1$ . Among 11, the place where the number to be set to is decided in a manner that a place having larger value than contents of the sequential number storing register 15 (Fig.4) and the first valid bit is "0" is found according to the instruction of  $L_2$  (that is, a valid bit), and the queue register number is set to there and the valid bit is also set to "1". The contents of 11 are sent to the sequential execution control device 10 through  $L_5$ ,  $L_{21}$ , and  $L_{22}$  to read out 8 and to control updating of 11. Computing results of 5, 6, and 7 are sent to 8 through  $L_{12}$ ,  $L_{14}$ , and  $L_{16}$ . At that time,  $L_{11}$ ,  $L_{13}$ , and  $L_{15}$  are set to "1" respectively. Where to be set among 8 is controlled by  $L_9$ . Setting to 8 is performed according to a queue register number of  $L_9$  and controlling of the sequential number. Controlling is performed at 10 according to states of  $L_6$ ,  $L_7$ , and  $L_8$ , and one of  $L_{17}$ ,  $L_{18}$ ,  $L_{19}$  is selected with controlling of  $L_{10}$ , then setting is performed on a general register, a floating point register and a storage writing register and so on (9) through  $L_{20}$ .

Fig.4 is a detailed drawing of the sequential execution control device 10. The role of 10 is explained with Fig.4.

In this figure, numerals 12, 13, 14, 18, 19, 20, 22, and 23 denote an AND gate respectively, 21 denotes an OR gate, 15 denotes a sequential number storage register, 16 denotes a circuit to update (+1) contents of 15, and 17 denotes a delay line. The sequential number and the valid bit are sent through  $L_{21}$  and  $L_{22}$ , and when AND is effected at 12,  $L_{23}$  becomes to be ON, and the queue register number is sent through  $L_5$ , and when AND is effected at 13,  $L_9$  becomes to be ON. On the other hand, when AND among  $L_9$ ,  $L_{21}$ , and  $L_{27}$  is effected at 14,  $L_{10}$  becomes to be ON, 8 performs reading. At that time, the sequential number (reading out an updated value by 16 from 15) and the queue register number are sent from 22 and 23 to 11 through  $L_3$  and  $L_4$ . When AND is not effected at 14,  $L_{10}$  becomes to be OFF for not starting 16. At that time,  $L_{28}$  becomes to be ON, and  $L_{29}$  is turned to ON delaying one

cycle from 17 for controlling 12. On the other hand,  $L_6$ ,  $L_7$ , and  $L_8$  are controlled with  $L_5$ , an AND condition for one of 18, 19, and 20 is effected, one of  $L_{24}$ ,  $L_{35}$ , and  $L_{26}$  becomes to be ON, and  $L_{27}$  is controlled through 21. This is for checking whether computing has been completed.

By placing a device above mentioned, in a computer having plural of instruction execution units, high speed processing of instruction words is expected with processing computing results independently and sequentially.

#### Brief Description of the Drawings

Fig.1 shows a structure of general digital computer.

Fig.2 is a drawing to show a format of a queue register and a computing execution control table according to the present invention.

Fig.3 is a block diagram of an instruction execution unit according to the present invention.

Fig.4 shows a sequential execution control device according to the present invention.

- 1: main memory device
- 2: instruction control unit
- 3: memory device control unit
- 4: instruction execution unit
- 8: queue register
- 11: computing execution control table
- 10: sequential execution control device